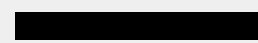


How Can Machine Learning Increase Students' exposure to Career Opportunities?



Use of NLP to increase exposure to Career Opportunities

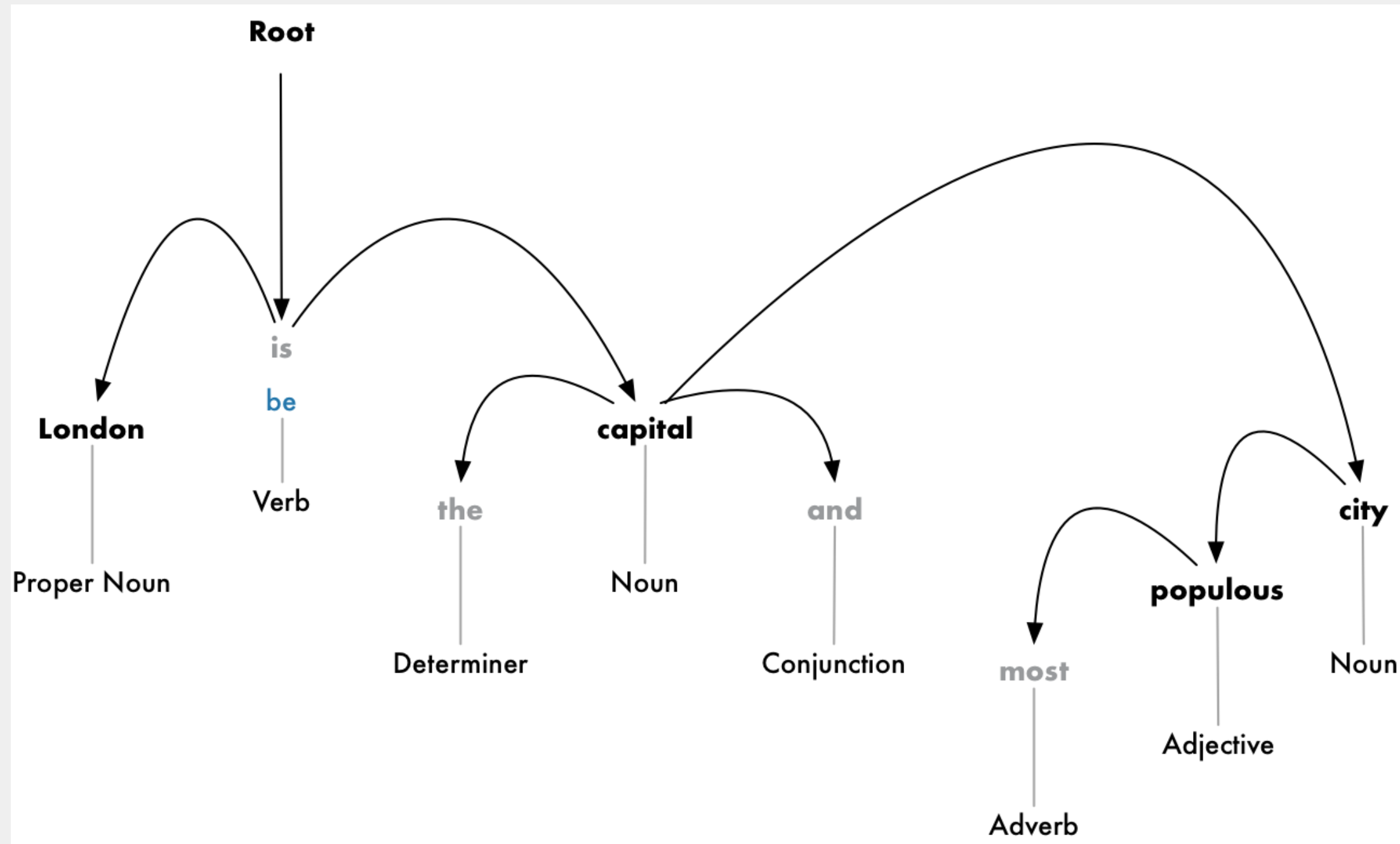
What is Natural Language Processing

NLP is a subfield of linguistics, computer science, and artificial intelligence concerned with the interactions between computers and human language, in particular how to program computers to process and analyze large amounts of natural language data.

The goal of Natural Language Processing

The goal is a computer capable of "understanding" the contents of documents, including the contextual nuances of the language within them. The technology can then accurately extract information and insights contained in the documents as well as categorize and organize the documents themselves.

How NLP works

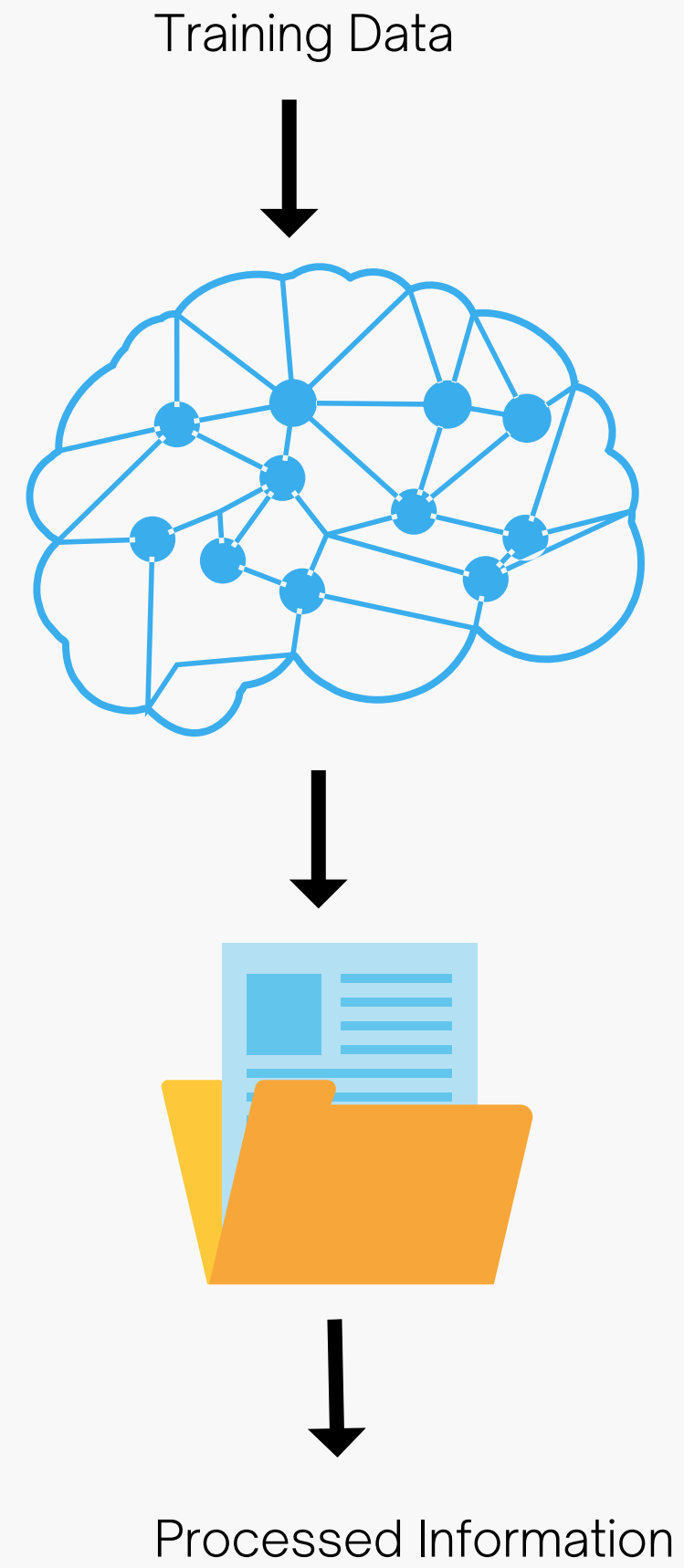


Building a pipeline

The idea is to break up your problem into very small pieces and then use machine learning to solve each smaller piece separately.



Machine Learning Algorithms and NLP



Problem Statement

The goal of Office of Career Development is to help as many students as possible achieve their career objectives yet there is no established system to help the office make decisions based on real time data.

Note that a 2016 Gallup poll showed that 61 percent of students who graduated between 2010 and 2016 said they had visited a career center at least once and 17 percent found it *very effective*

In this research, I will be investigating how to use machine learning to increase the effectiveness of the Office of Career Development in sending career opportunities and for students to increase their productivity while finding career opportunities.

Methodology

- Using an experimental approach
- Have a set of students using an application and others without the application and find out the difference in outreach of such career opportunities

1. Creating a test application

Application Framework: Django

Reason: Lightweight and fast

Purpose: Test keyword extraction and rule-based matching

Requirements: Spacy(free open-source library for Natural Language Processing in Python), Microsoft Graph

2. Retrieving Emails

```
"@odata.context": "https://graph.microsoft.com/v1.0/$metadata#users('b9b93453-8598-428e-b723-dc49e4a1476c')/messages",
"@odata.nextLink": "https://graph.microsoft.com/v1.0/me/messages?$skip=10",
"value": [
  {
    "@odata.type": "#microsoft.graph.eventMessageRequest",
    "@odata.etag": "W/\"CwAAABYAAAC/pclfsKGFQaQkHSL8jmXTAAK0b+jJ\"",
    "id":
    "AAMkADA5ZTg2ZmQ3LTEwMGQtNDk5Zi040DA3LTRmMWNkYmVjYjAzNABGAAAAAARD1vhf5kWTIY8nVk0Q34PBwC-pclfsKGFQaQkHSL8jmXTAAABYAAAC-pclfsKGFQaQkHSL8jmXTAAK0b+jJ",
    "createdDateTime": "2022-03-26T20:38:21Z",
    "lastModifiedDateTime": "2022-03-26T20:43:22Z",
    "changeKey": "CwAAABYAAAC/pclfsKGFQaQkHSL8jmXTAAK0b+jJ",
    "categories": [],
    "receivedDateTime": "2022-03-26T20:38:21Z",
    "sentDateTime": "2022-03-26T20:38:17Z",
    "hasAttachments": false,
    "internetMessageId": "<MN2PR11MB4349FD409DD68576A8D6F4C3A01B9@MN2PR11MB4349.namprd11.prod.outlook.com>",
    "subject": "[RSVP] MacVincent Agha-Oko Thesis Presentation",
    "bodyPreview": "Good afternoon,I am writing to formally invite you to my senior thesis presentation. My research project is titled Simulator to Real-World Transfer of Robot Learning Models. In my thesis, I explore the possibility of transferring a model trained in a ",
    "importance": "normal",
    "parentFolderId":
    "AAMkADA5ZTg2ZmQ3LTEwMGQtNDk5Zi040DA3LTRmMWNkYmVjYjAzNAAUAAAAAARD1vhf5kWTIY8nVk0Q34PAQC-pclfsKGFQaQkHSL8jmXTAAABYAAAC-pclfsKGFQaQkHSL8jmXTAAK0b+jJ",
    "conversationId": "AAQkADA5ZTg2ZmQ3LTEwMGQtNDk5Zi040DA3LTRmMWNkYmVjYjAzNAAQAGSgi-UnA05Fizl7mQtxn0w=",
    "conversationIndex": "A0HY0VFs7KCL9ScDTkWL0Xu7C3GfTA="
```

- User gives permission for the application to access emails through graph
- Emails are received in JSON format with various other datapoints

2. Cleaning data

- process of fixing or removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset

```
args = [
    ("\\r\\n", " "), #Removes \r\n
    ("(http|ftp|https):\/\/([\w_-]+(?:\.[\w_-]+)+)([\w.,@?^=%&\/~+#-]*[\w@?^=%&\/~+#-])?", " "), #
    ("< >", " "), #removes < >
    ("\\[ \\]", " "), # Removes [ ]
    ("\\[.*?\\]", " "), #Removes text within [ ]
    ("<.*?\\>", " "), #Removes text within <>
    ("_", " "), #Removes dashes
    ("\\*", " "), #Removes dashes
    ("CAUTION: This email originated from outside the organization. Do not click links or open atta",
    " "),
    ("[ ]{2,}", " "), #Removes every empty space whose len is greater than 1
]
def clean_data(args, text):
    # Create a regular expression from the tuple
    for old, new in args:
        text = re.sub(old, new, text)
    return text
```

Token-based matching

- spaCy features a rule-matching engine, the Matcher, that operates over tokens, similar to regular expressions

```
def claflin():
    matcher = Matcher(nlp.vocab)
    #Needs improvement
    pattern = [
        [{"LOWER": "apply"}], [{"LOWER": "opportunity"}], [{"LOWER": "internship"}], [{"LOWER": "intern"}], [{"
    ]
    finalData = {"mimetype": "application/json"}
    } #Holds cleaned data whose senders are from Claflin and whose body contains the above keywords
    arr = []
    for mail in data:
        subdata = {}
        claflinMail = re.search('@claflin\\.edu$', mail['sender']['emailAddress']['address'])
        if claflinMail:
            #Save links in a separate map for later use
            subdata['Links'] = re.search("((http|ftp|https):\/\/([\w_-]+(?:([\w_-]+)+))(\.[\w.,@?^=%&:/~+#-]*[\w
            body = clean_data(args, mail['body']['content'])
            doc = nlp(body)
            #text1= spacy.NER(bod)
            entities = [(ent.text, ent.label_) for ent in doc.ents]
            matcher.add("OpportunityKeys", pattern)
            matches = matcher(doc)
            for match_id in matches:
                if match_id and (mail['id'] not in finalData):
                    document = nlp(body)
                    entities = [(ent.text, ent.label_) for ent in document.ents]
                    finalData[str(mail["id"])] = {"body": body, "subject": mail["subject"], "sender": mail["send
    return finalData
```


Results and Application

- Out of the 1000 emails extracted, 153 were career related, this represents 15.3 percent of emails.
- Not found a clear metric to find the accuracy of my model
- This can be used in creating email filters to easily sift through career related opportunities.
- A full scale application can be built based on this to help instructors gauge analyze how students interact with such emails

Problems

- Email data is very sensitive and users may not oblige to share such data
- Even with users agreeing to share data, storage of such information requires high levels of data security.
- The problem can be solved without NLP, have students share opportunities instead of reliance to emails.

Solutions

- Using a smaller sample size
- Using Blockchain technology to implement the application

Near Future Goals

- Creating a full scale application
- Creating a blockchain application

References

- Marken, Z. A. and S. (2022, April 6). One in six U.S. grads say career services was very helpful. Gallup.com. Retrieved April 17, 2022, from <https://news.gallup.com/poll/199307/one-six-grads-say-career-services-helpful.aspx>